

Лабораторная работа № 3
по дисциплине «Интеллектуальные информационные системы»
«Создание информационной среды сопровождения жизненного цикла
интеллектуальных систем»

1. Изучить функциональные возможности языка логического программирования Prolog.
2. Проанализировать свойства персональной рабочей платформы информатизации.
3. Изучить функциональные возможности системы Visual Prolog.
4. Найти в глобальной сети Internet систему Visual Prolog.
5. Выбрать определённый вариант системы Visual Prolog, соответствующий состоянию персональной рабочей платформы информатизации. Примеры регистрации реализации выбора приводятся на рис. П1.1 и рис. П1.2 Приложения 1.
6. Установить систему Visual Prolog на рабочую платформу информатизации с регистрацией выполняемых действий в виде копий экранов.
7. Развернуть систему Visual Prolog и ознакомиться с основными функциональными возможностями среды разработки (Приложение 2, Приложение 3).
8. Построить базу данных относительно родства двух человек согласно примеру с рис. 1.

Родитель (Илья, Марина)
Родитель (Марина, Ира)
Родитель (Елена, Иван)
Родитель (Николай, Ира)
Родитель (Ольга, Алексей)
Родитель (Марина, Саша)
Родитель (Сергей, Иван)

Рисунок 1 – Пример базы знаний

9. Составить запрос для определения правильности утверждения: Марина является родителем Саши.
10. Реализовать составленный согласно п.9 запрос в системе Visual Prolog и создать копии реализации. Добиться успешной реализации составленного запроса.
11. Составить запрос для определения правильности утверждения: Алексей является родителем Ольги.
12. Реализовать составленный согласно п.11 запрос в системе Visual Prolog и создать копии реализации. Добиться успешной реализации составленного запроса.
13. Составить запрос для установления факта: кто является ребёнком

Николая?

14. Реализовать составленный согласно п.13 запрос в системе Visual Prolog и создать копии реализации. Добиться успешной реализации составленного запроса.

15. Составить запрос: кто является родителями Ивана?

16. Реализовать составленный согласно п.15 запрос в системе Visual Prolog и создать копии реализации. Добиться успешной реализации составленного запроса.

17. Составить запрос для определения всех родителей и их детей.

18. Реализовать составленный согласно п.17 запрос в системе Visual Prolog и создать копии реализации. Добиться успешной реализации составленного запроса.

19. Проанализировать результаты выполненных экспериментов.

20. Примеры результатов успешной деятельности по установке пакета приведены в Приложении.

21. Составить отчёт по выполненной лабораторной работе. В отчёт включить: цель работы, задание, копии выполненных действий с комментариями, представляющими их содержание, и вывод.

22. Представить отчёт о выполненной работе преподавателю и защитить его.

23. Исправить отчёт по выполненной работе в соответствии с замечаниями преподавателя, распечатать и подписать титульный лист отчёта, сдать преподавателю электронную копию отчёта и подписанный титульный лист.

Пример инсталляции системы Visual Prolog

Для примера инсталляция системы Visual Prolog производилась на платформу информатизации Samsung R518 с установленной операционной системой Windows 7, имеющей разрядность 32 бита. Характеристики используемой платформы представляются на рис. П1.1.

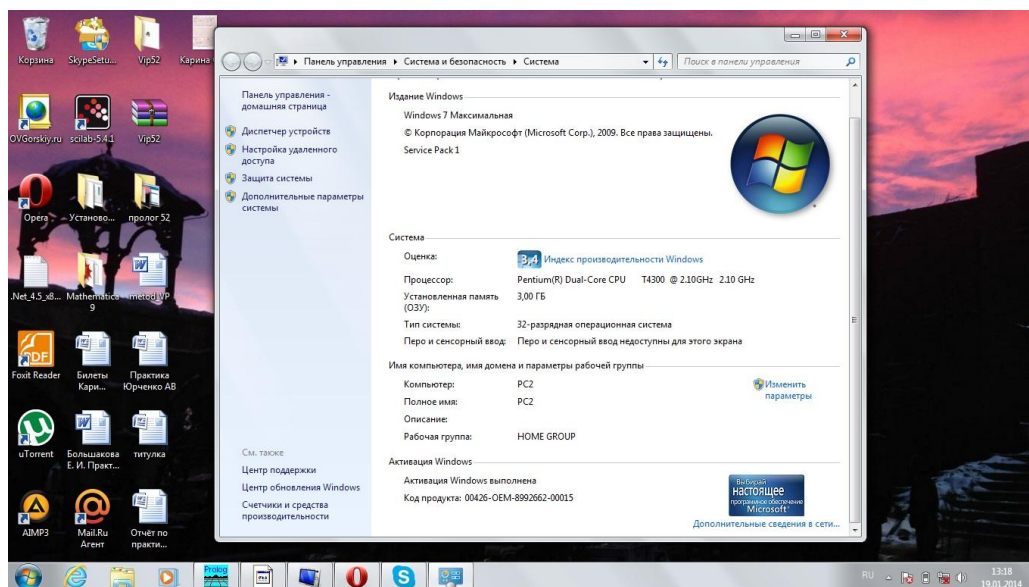


Рисунок П1.1 – Характеристики платформы информатизации

Выбор подходящей состоянию рабочей платформы версии Visual Prolog в глобальной сети Internet демонстрируется на рис. П1.2.

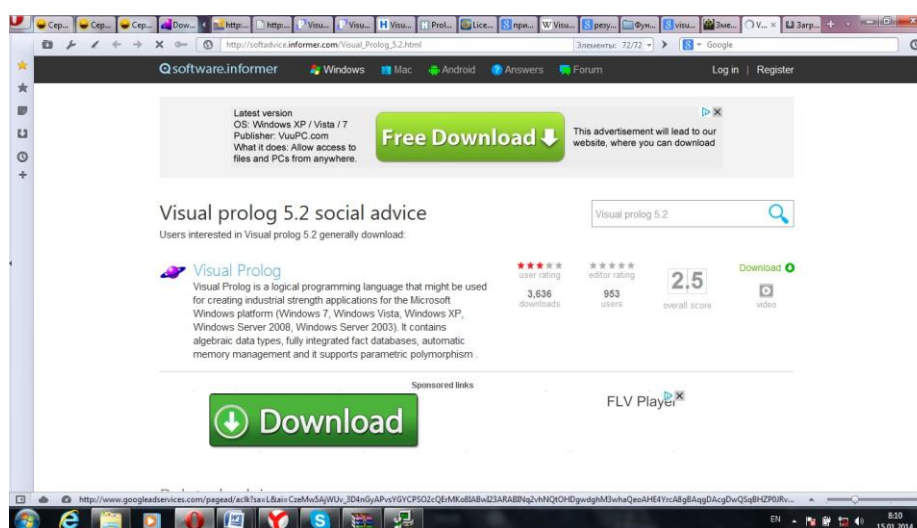


Рисунок П1.2 – Выбор версии Visual Prolog

Сохранение установочного пакета отображается на рис. П1.3.

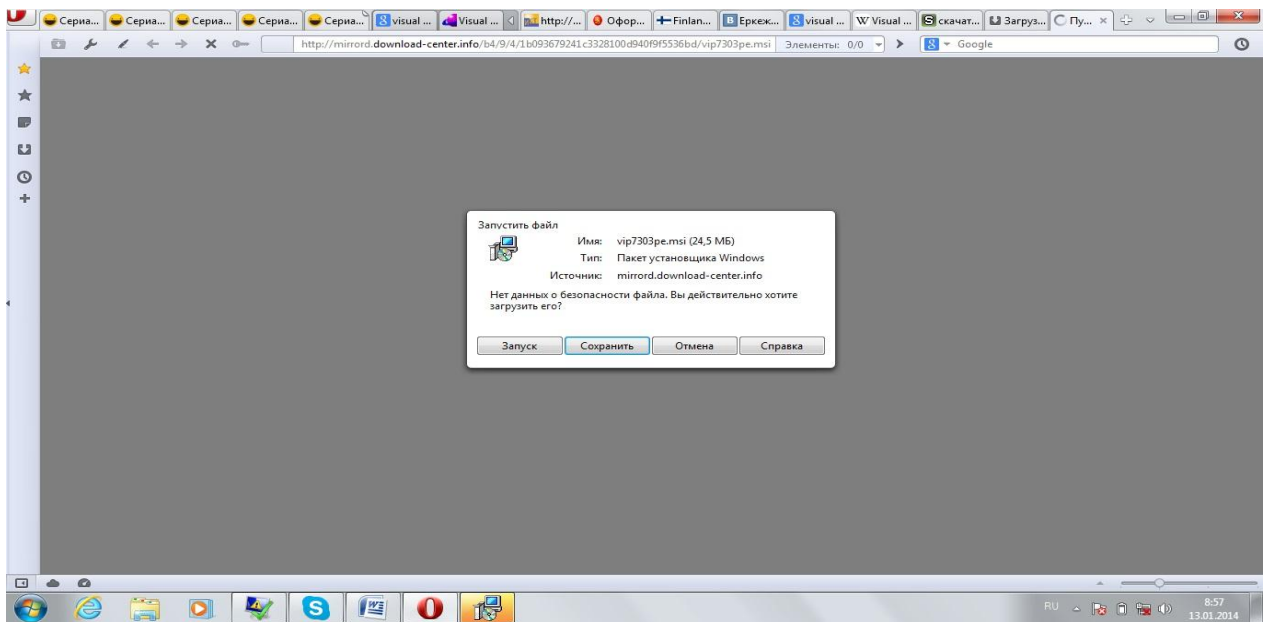


Рисунок П1.3 – Сохранение установочного файла

Зарегистрированные копии успешных действий по установке выбранной системы на рабочую платформу информатизации приводятся на рис. П1.4, рис. П1.5, рис. П1.6, рис. П1.7, рис. П1.8, рис. П1.9.

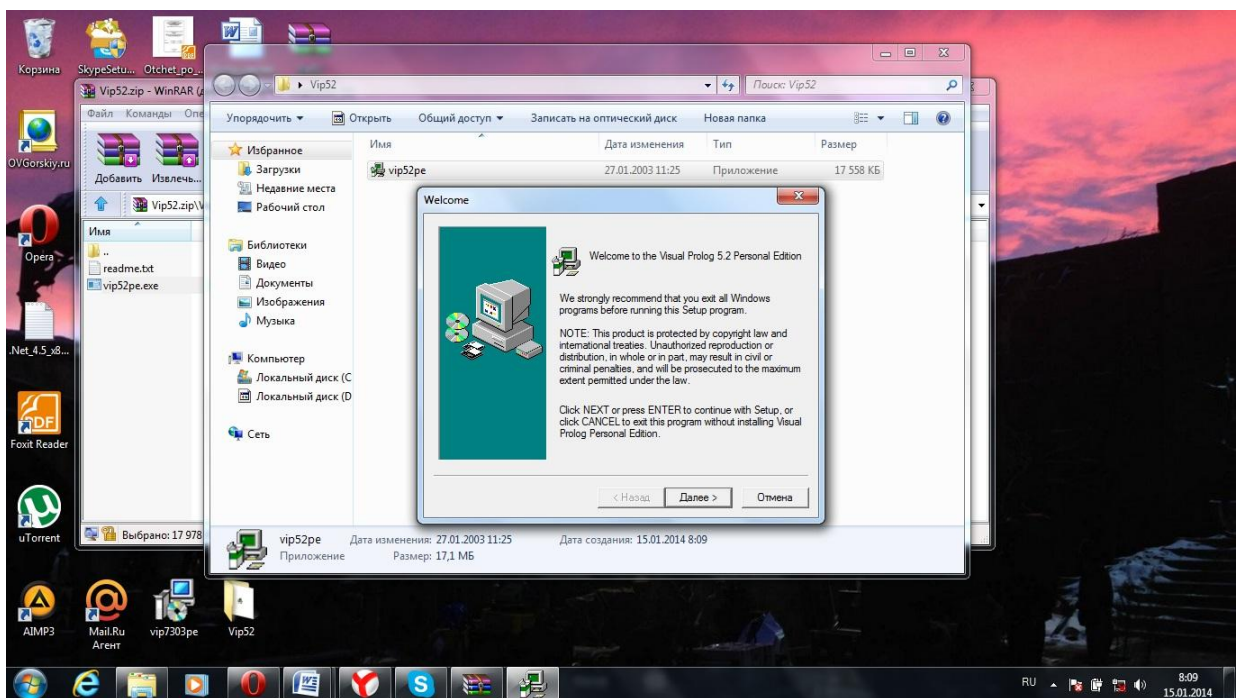


Рисунок П1.4 – Стартовое окно запуска установки Visual Prolog

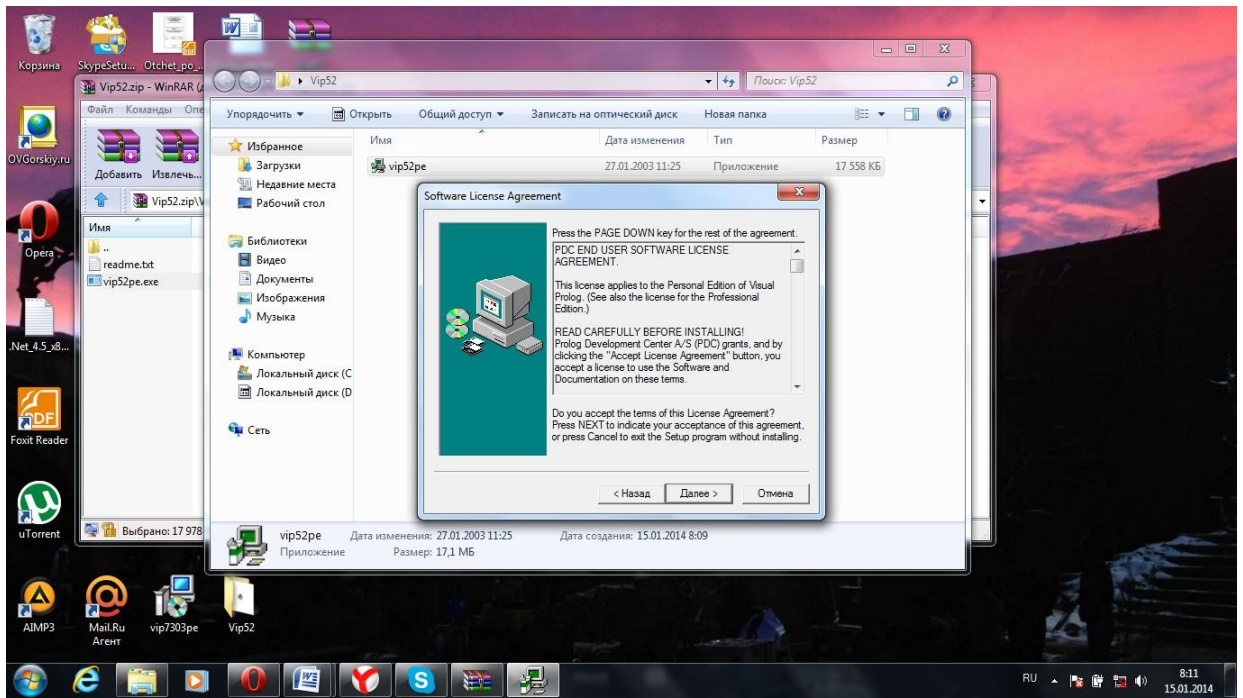


Рисунок П1.5 – Лицензионное соглашение на использование Visual Prolog версии 5.2

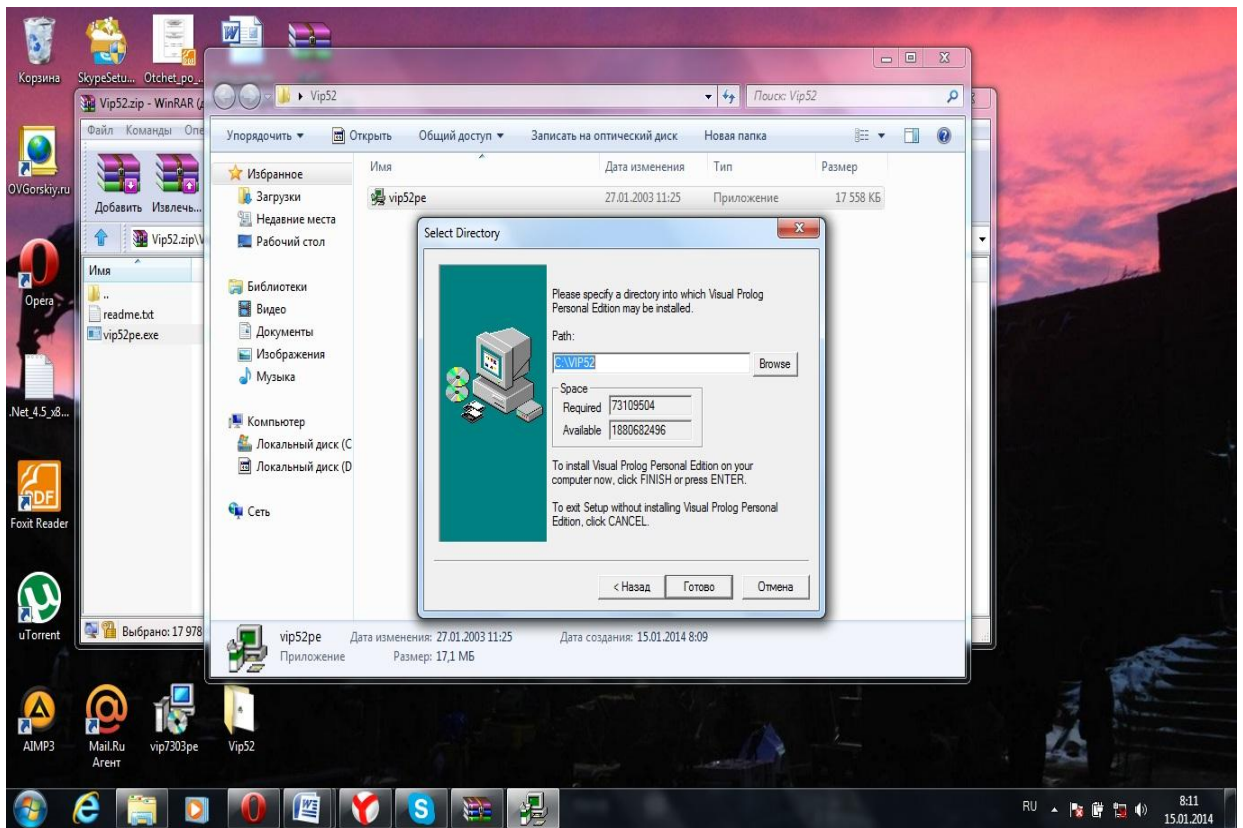


Рисунок П1.6 – Выбор месторасположения

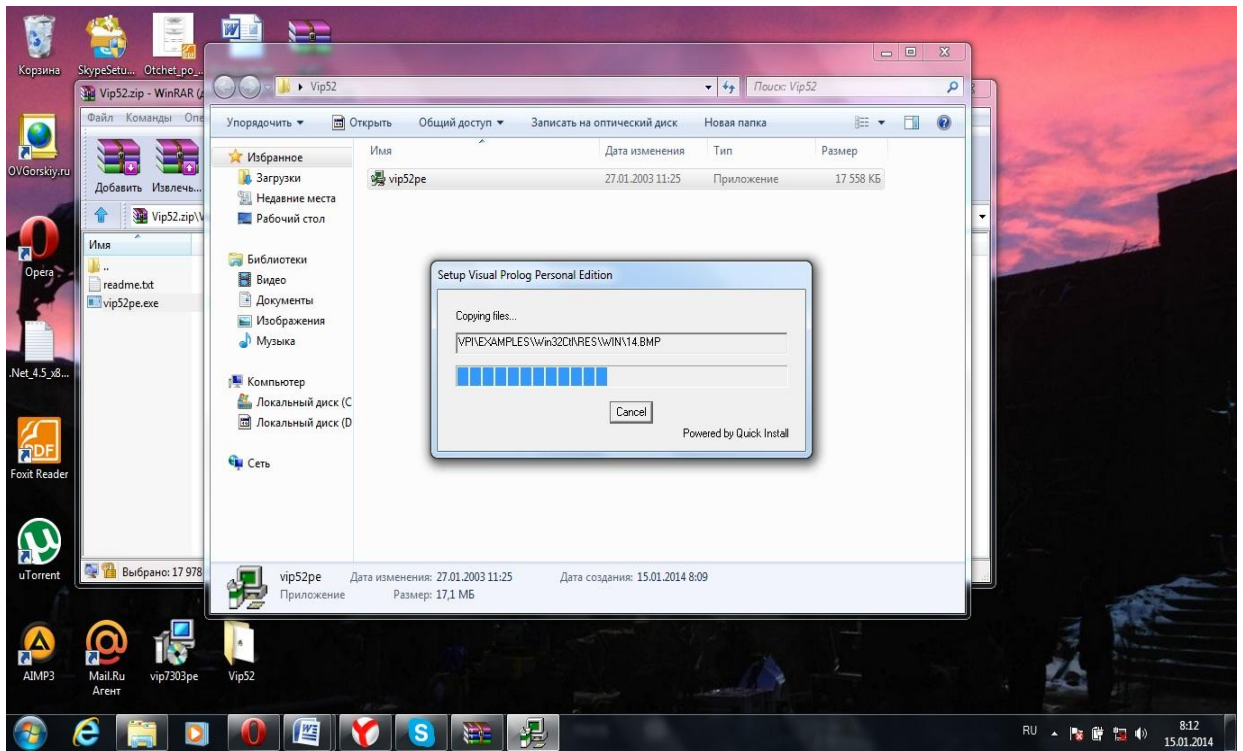


Рисунок П1.7 – Процесс установки

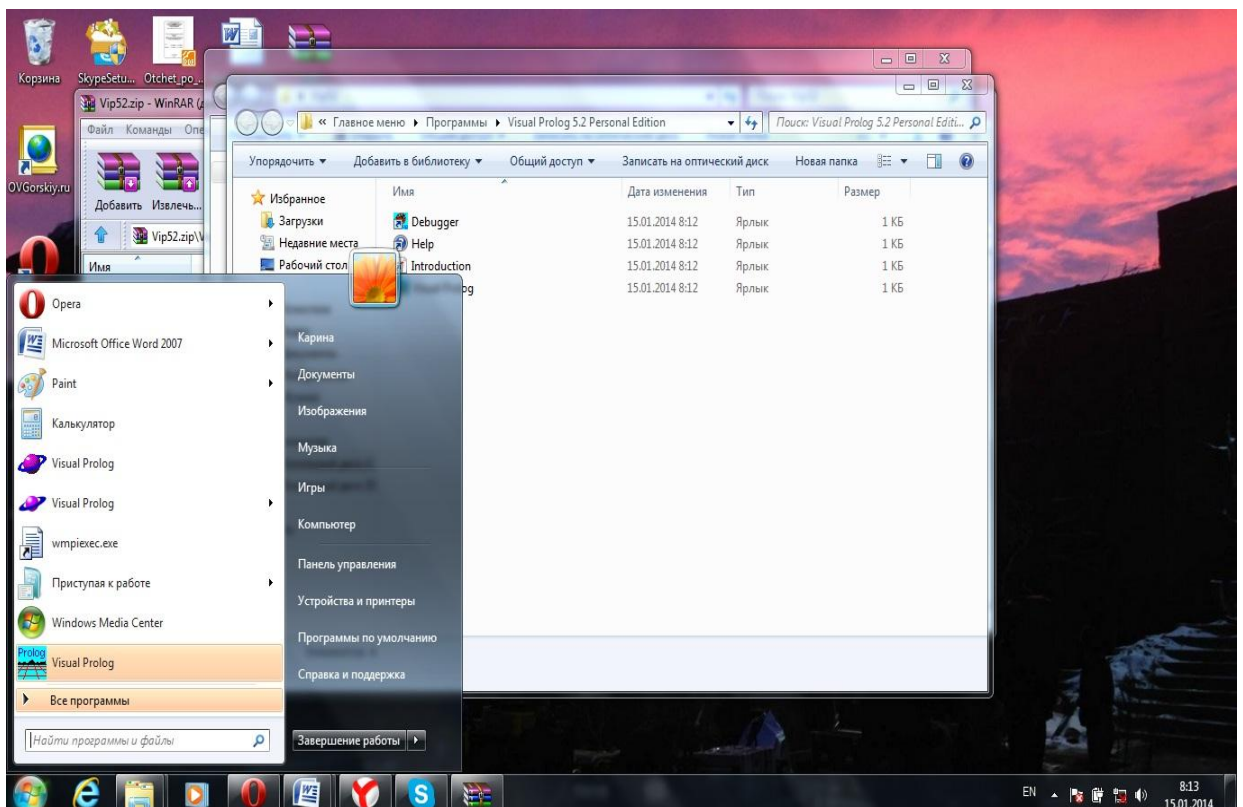


Рисунок П1.8 – Проверка в меню «ПУСК»

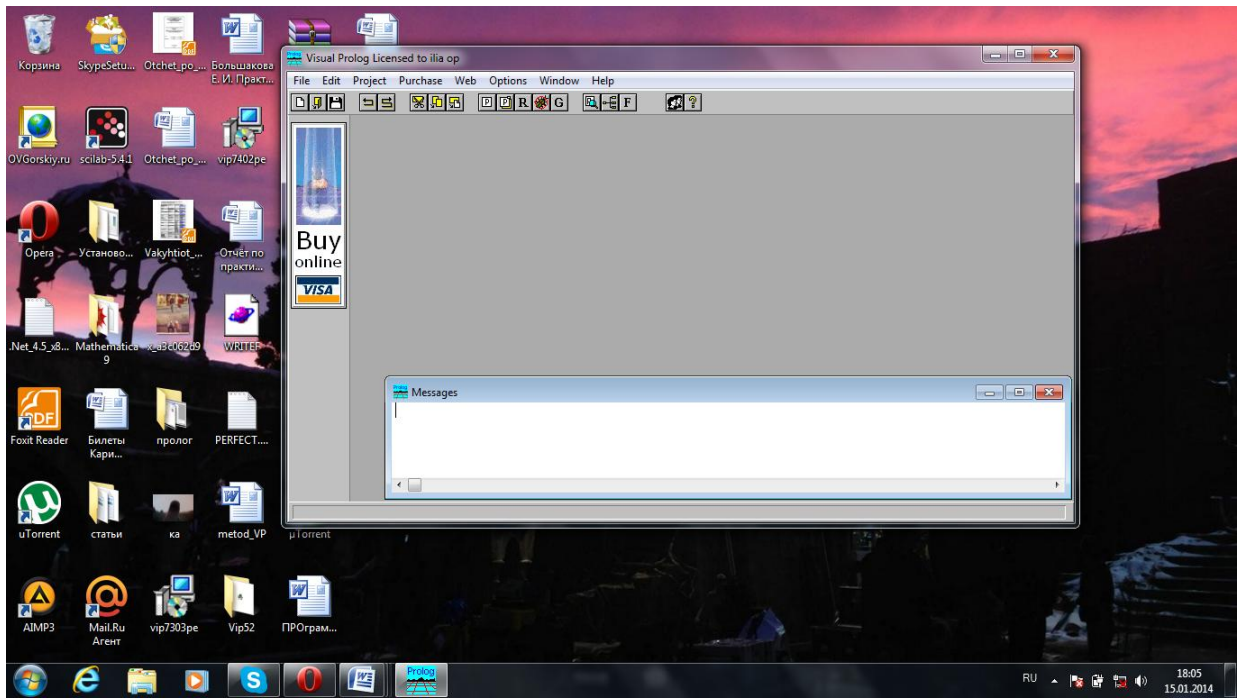


Рисунок П1.9 – Рабочее окно системы

Интерактивная среда визуальной разработки

В систему Visual Prolog входит интерактивная среда визуальной разработки (VDE – Visual Develop Environment), которая включает текстовый и различные графические редакторы, инструментальные средства генерации кода, конструирующие управляющую логику (Experts), а также являющийся расширением языка интерфейс визуального программирования (VPI – Visual Programming Interface), Пролог-компилятор, набор различных подключаемых файлов и библиотек, редактор связей, файлы, содержащие примеры и помощь.

Visual Prolog поддерживается различными ОС, в том числе MS-DOS PharLap-Extended DOS, всеми версиями Windows, 16- и 32-битовыми целевыми платформами OS/2, а также некоторыми другими системами, требующими графического пользовательского интерфейса.

В зависимости от выбранного интерфейса разработчику обеспечивается доступ к множеству генераторов кода (Code Expert), всевозможным ресурсным редакторам и особым дополнительным VPI-предикатам, определениям и библиотекам. Ресурсные редакторы применяются для создания, компоновки и редактирования окон, диалогов, меню, панелей инструментов, строк помощи, строковых таблиц, ярлыков, курсоров, битовых карт и оперативной помощи. Генераторы кода на основе подобных структур создают необходимый первичный Prolog-код. В результате появляется первичный код («скелет»), готовый для компиляции, редактирования связей и выполнения.

В интерфейс системы Visual Prolog включается: главное меню, панель инструментов, окно проекта. Если во время последнего использования системы Visual Prolog там был открытый проект, то система автоматически вновь откроет этот проект.

На рис. П2.1 отображается внешний вид среды Visual Prolog после запуска. В окне проекта отображаются модули открытого проекта route.prj: karta.pro, route.pro, VPITools.pro.

Левая панель кнопок в окне проекта позволяет выбирать нужный компонент проекта: модуль, окно, меню и т.д. С помощью кнопок правой панели выбранный компонент можно редактировать (кнопка Edit), удалять (кнопка Delete), а также добавлять новый (кнопка New).

Пункт меню **File** содержит команды для работы с файлами. Чтобы создавать новое окно редактирования, можно использовать команду File | New. Эта команда создаст новое окно редактора с заголовком «NONAME».

В меню **Edit** представлены команды, позволяющие редактировать текст программы. Встроенный редактор системы по интерфейсу похож на обычный текстовый редактор. Можно производить вырезку, копирование и вставку текста, операции Отмена/Восстановление, которые можно активизировать из меню Edit. Также меню Edit показывает «горячие клавиши», связанные с соответствующими действиями.

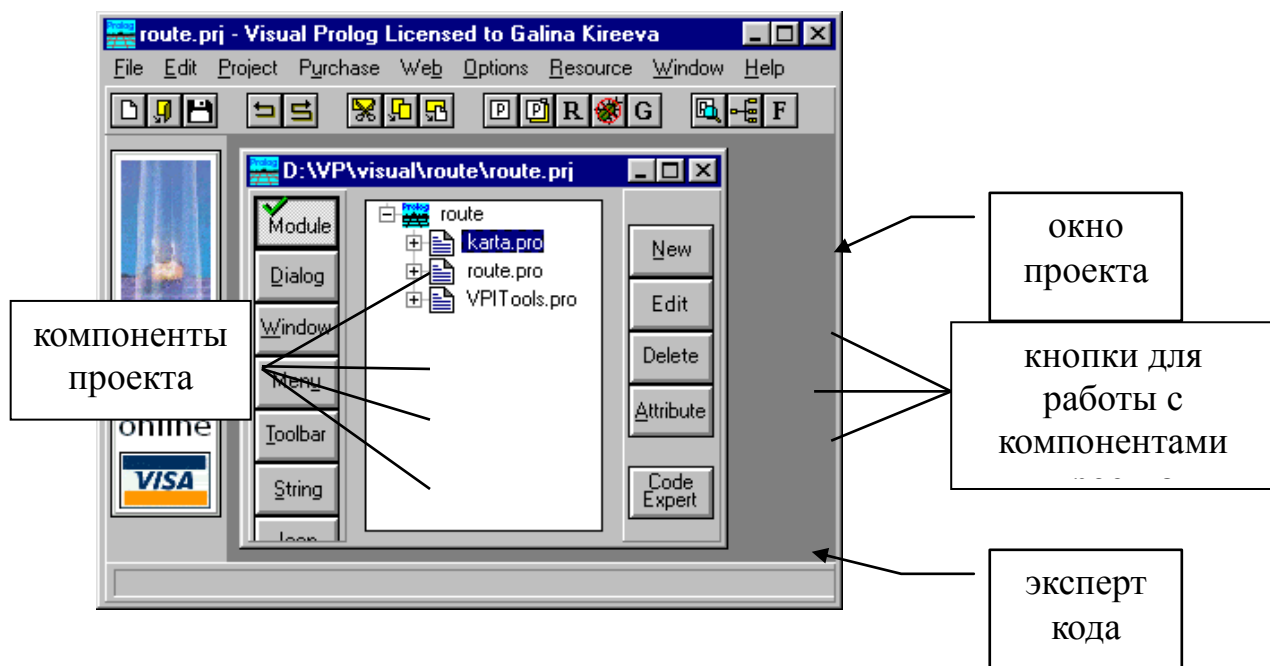


Рисунок П2.1 – и Среда разработки Visual Prolog

Пункт меню **Project** содержит команды для работы с проектом: создать новый, открыть, запустить и т.д. Запуск проекта на исполнение выполняется нажатием кнопки <R> на панели инструментов (или F9, или с помощью команд меню Project | Run).

Команды меню **Options** позволяют выполнять настройку проекта, устанавливать необходимые параметры.

Основные сведения о разработке и тестировании программ на языке ПРОЛОГ

Программа на языке ПРОЛОГ состоит из предложений, которые могут быть фактами, правилами или запросами. Как правило, программа состоит из четырех разделов.

DOMAINS – секция описания доменов (типов). Секция применяется, если в программе используются нестандартные домены.

PREDICATES – секция описания предикатов. Секция применяется, если в программе используются нестандартные предикаты.

CLAUSES – секция предложений. Именно в этой секции записываются предложения: факты и правила вывода.

GOAL – секция цели. В этой секции записывается запрос.

Среда Visual Prolog позволяет протестировать программу без создания проекта. Для этого используется утилита Test Goal. Достаточно создать новый файл, набрать текст программы и активизировать Test Goal нажатием кнопки <G> на панели инструментов. Автономно исполняемый файл при этом не создается. Утилита Test Goal компилирует только тот код, который определен в активном окне редактора (код в других открытых окнах или модулях проектов, если они есть, игнорируются). Test Goal находит *все* возможные решения задачи и автоматически выводит значения *всех* переменных.

Предикат write. Создание проекта позволяет протестировать пример как автономную исполняемую программу. После запуска проекта на исполнение создается exe-файл, работа которого завершается после *первого* решения, удовлетворяющего решению задачи. Запуск программы в этом режиме не обеспечивает автоматический вывод значений переменных, поэтому необходимо использовать стандартный предикат вывода **write**.

Предикат fail. Fail – это тождественно-ложный предикат, искусственно создающий ситуацию неуспеха. После выполнения этого предиката управление передается в точку отката и поиск продолжается. Использование предиката fail позволяет найти все решения задачи.

Чтобы ограничить пространство поиска и прервать поиск решений при выполнении какого-либо условия, используется предикат *отсечения* (обозначается !), Однажды пройдя через отсечение, невозможно вернуться назад, т.к. этот предикат является тождественно-истинным. Процесс может только перейти к следующей подцели, если такая имеется.

Пролог не предназначен для решения вычислительных задач, но его возможности вычислений аналогичны соответствующим возможностям таких языков программирования как Basic, C, Pascal.

В языке Пролог имеется ряд встроенных функций для вычисления арифметических выражений, ряд из которых перечислен в табл. ПЗ.1.

Математические операции и функции в Прологе

$X + Y$	Сумма X и Y
$X - Y$	Разность X и Y
$X * Y$	Произведение X и Y
X / Y	Деление X на Y
$X \bmod Y$	Остаток от деления X на Y
$X \operatorname{div} Y$	Целочисленное деление X на Y
$\operatorname{abs}(X)$	Абсолютная величина числа X
$\operatorname{sqrt}(X)$	Квадратный корень из X
$\operatorname{random}(X)$	Случайное число в диапазоне от 0 до 1
$\operatorname{random}(\operatorname{Int}, X)$	Случайное целое число в диапазоне от 0 до Int
$\operatorname{round}(X)$	Округление X
$\operatorname{trunc}(X)$	Целая часть X
$\sin(X)$	Синус X
$\cos(X)$	Косинус X
$\operatorname{arctan}(X)$	Арктангенс X
$\tan(X)$	Тангенс X
$\ln(X)$	Натуральный логарифм X
$\lg(X)$	Логарифм X по основанию 10